WaterlooClarke at the TREC 2021 Conversational Assistant Track

Xinyi Yan, Charles L. A. Clarke, Negar Arabzadeh

November 6, 2021

1 Introduction

For TREC 2021, the WaterlooClarke group submitted three runs to TREC Conversational Assistance Track (CAsT):

- clarke-auto
- clarke-cc
- clarke-manual

The three runs were based on raw utterances, canonical response, and manually rewritten utterances respectively. This report describes the generation and the results of each of these runs.

The overall approach consists of three steps: 1) query reformulation, 2) passage retrieval, and 3) passage reranking. We did not apply the query reformulation step for the **clarke-manual** run as manually rewritten utterances were used. In order to improve our performance, this year we focused our effort on maximizing the total system recall at the first stage by employing both dense and sparse retrievers. Research has shown that sparse retrievers and dense retrievers can retrieve complementary information [3]. We merged the retrieved passages into a single pool, then reranked this pool using a two-stage reranking pipeline with monoT5 and duoT5 [4]. In the next section, we will explain the details of our methodology.

2 Methodology

2.1 Query Reformulation

Similar to last year, the year 3 conversations are multi-turn, contextually dependent and informally expressed. To incorporate essential information from previous context and resolve issues such as co-reference and word omissions, we trained a T5 model on the QReCC dataset[1] to reformulate raw queries.

In general, query reformulation task is defined as follows. Given a conversational context $c = [u_0, p_0, u_1, p_1, ..., u_{i-1}, p_{i-1}]$, and current query u_i, the goal is to use the context to rewrite u_i as $\hat{u_i}$ where $\hat{u_i}$ is context-independent and would be consumed directly by retrievers.

Based on experiments on the TREC CAsT 2019 and 2020 datasets, we chose to use previous rewritten utterances $[\hat{u_0}, \hat{u_1}, ..., \hat{u_{i-1}}]$ and the last system response p_{i-1} as context to rewrite u_i:

$\hat{u_i} = T5(\hat{u_0}, \hat{u_1}, ..., \hat{u_{i-1}}, p_{i-1}, u_i)$

The **clarke-auto** run and the **clarke-cc** run both used query reformulation technique. The difference is the choice of the system response passage p_{i-1}. In **clarke-auto**, we used top five passages retrieved purely by our system as context to rewrite next queries, whereas in **clarke-cc** the provided canonical results were used. While the T5 re-writer works surprisingly well on the CAsT 2019 and 2020 datasets, we do observe that the model more often fails to find the omitted information or detect topic shift in CAsT 2021, which suggests CAsT 2021 tasks are harder than before.

2.2 Passage Retrieval

This year we utilized the BERT-based dense retriever ANCE [5] and a tuned BM25 with pseudo-relevance feedback to retrieve passages for a set of rewritten queries at very first stage. BM25 parameters were tuned to maximize recall@1000 over the 2019 and 2020 tasks using the manually formulated questions from those years. The pseudo-relevance feedback step executed queries over both the target corpus and the much larger C4 corpus developed to train T5[1].Then we merged the results into a single pool that would be reranked in the next stage. Our experiments on the CAsT 2019 and 2020 datasets have shown that broadening the pool of retrieved passages can lead to a significant improvement in the performance [2].

2.3 Passage Reranking

On all the three runs, we applied a pointwise reranker monoT5, followed by a pairwise reranker duoT5 [4]. According to recent research and experiments in passage ranking tasks such as MSMARCO, neural rerankers can significantly improve the final performance.

3 Results

Table 1 compare our three runs based on ndcg@3, ndcg@5, recall@500 and recall@R, where the former three are the main measures[2] used by the 2021 CAsT track, and R is the number of relevant documents for that query. Not surprisingly, the use of manual rewrites led to enhanced performance, which suggested

[1] `https://huggingface.co/datasets/allenai/c4`

[2] Evaluation is based on documents

that there is room left for improvements in the query reformulation stage. In addition, we investigated how much the use of different first stage retrievers would contribute to the final scores. Table 2 shows the results when manually resolved queries were used. We repeated the same analysis for canonical runs and raw runs, which were demonstrated in Table 3 and Table 4. In general, adding pseudo-relevance feedback and combining different first stage retrievers improved the system performance and further implied that dense retrievers and sparse retrievers could be combined together to achieve better performance.

Run	ndcg@3	ndcg@5	recall@500	recall@R
clarke-auto	0.3753	0.3685	0.6760	0.2576
clarke-cc	0.5137	0.5107	0.8303	0.3549
clarke-manual	0.6440	0.6382	0.8894	0.4537

Table 1: Comparing the results of our three submitted runs

First-stage Retrievers	ndcg@3	ndcg@5	recall@500	recall@R
Baseline BM25	0.5952	0.5882	0.7459	0.4210
BM25	**0.6464**	0.6381	0.8083	0.4530
BM25 + C4	0.6409	0.6375	0.8378	**0.4545**
ANCE	0.6284	0.6157	0.7843	0.4400
BM25 + C4 + ANCE	0.6440	**0.6382**	**0.8894**	0.4537

Table 2: Comparing the results of manual runs when different first-stage retrievers were used

First-stage Retrievers	ndcg@3	ndcg@5	recall@500	recall@R
Baseline BM25	0.4357	0.4265	0.6168	0.3067
BM25	**0.5166**	**0.5073**	0.7933	0.3482
BM25 + C4	0.5076	0.5007	0.8015	0.3481
ANCE	0.5105	0.5031	0.7650	0.3461
BM25 + C4 + ANCE	0.5083	0.4996	**0.8254**	**0.3496**

Table 3: Comparing the results of canonical runs when different first-stage retrievers were used

4 Conclusion

In TREC CAsT 2021, our team experimented with a tuned BM25 along with the dense retriever ANCE to expand the pool of retrieved documents at very first stage using a set of rewritten queries. The results have shown that our query reformulation methods can be further improved, possible future directions including detecting topic shifts in the conversations and better resolving co-references. In addition, we are interested in studying how to make use of hybrid retrievers more smartly, one example is to select between different retrievers

First-stage Retrievers	ndcg@3	ndcg@5	recall@500	recall@R
Baseline BM25	N/A	N/A	N/A	N/A
BM25	0.3708	0.3578	0.5798	0.2516
BM25 + C4	0.3553	0.3471	0.5739	0.2377
ANCE	0.3724	0.3674	0.5975	0.2490
BM25 + C4 + ANCE	**0.3753**	**0.3685**	**0.6760**	**0.2576**

Table 4: Comparing the results of raw runs when different first-stage retrievers were used

based on the query alone [3]. We look forward to participating in TREC CAsT 2022.

References

[1] Raviteja Anantha et al. *Open-Domain Question Answering Goes Conversational via Question Rewriting.* 2021. arXiv: `2010.04898 [cs.IR]`.

[2] Negar Arabzadeh and Charles L. A. Clarke. *WaterlooClarke at the Trec 2020 Conversational Assistant Track.* 2020.

[3] Negar Arabzadeh, Xinyi Yan, and Charles L. A. Clarke. *Predicting Efficiency/Effectiveness Trade-offs for Dense vs. Sparse Retrieval Strategy Selection.* 2021. arXiv: `2109.10739 [cs.IR]`.

[4] Ronak Pradeep, Rodrigo Nogueira, and Jimmy Lin. *The Expando-Mono-Duo Design Pattern for Text Ranking with Pretrained Sequence-to-Sequence Models.* 2021. arXiv: `2101.05667 [cs.IR]`.

[5] Lee Xiong et al. *Approximate Nearest Neighbor Negative Contrastive Learning for Dense Text Retrieval.* 2020. arXiv: `2007.00808 [cs.IR]`.